IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

This is a U.S. Patent Application for:

TITLE: DISTRIBUTED VOICE CONFERENCING

Inventor #1: Marcelo G. Oliveira
Address: 60 Descanso Drive #2219, San Jose CA 95134
Citizenship: Brazil

Inventor #2: Kiran Hassan
Address: 3765 Tamarack Ln #109, Santa Clara, CA 95051
Citizenship: India

DISTRIBUTED VOICE CONFERENCING

FIELD OF THE INVENTION

[001]   The present invention relates to voice conferencing. In particular, the present invention relates to expanding a conference over multiple media processors to efficiently extend the conferencing capabilities of a voice conferencing system.

BACKGROUND OF THE INVENTION

[002]   Effective communication is critical for successful business. The desire to enhance communication, in conjunction with incredible advances in processing technology, have lead to new and effective communication systems for businesses. For example, traditional data-only networks have now merged with traditional voice-only networks to form sophisticated hybrid Internet Protocol (IP) Telephone systems. The cost and performance benefits associated with IP Telephone systems has lead to their successful implementation in hundreds of companies.

[003]   One popular service now offered over IP Telephony systems is the voice conference. In a voice conference, multiple participants engage in discussions through the support of the IP Telephone backbone. The participants may be located virtually anywhere, with the backbone seamlessly connecting the participants as if they were in the same conference room.

[004]   In the past, the IP Telephony system assigned a single media processor to each voice conference. The assigned media processor handled the entire data flow generated by all participants in the voice conference. However, because the media processor had limited computational capabilities and memory resources, the media processor could only process a limited number of voice channels. Thus, additional individuals simply could not participate in a voice conference when the media processor channel limits had been reached.

1

[005] Depending on the resources available to the media processor, and the number of conference participants, a single media processor sometimes handled multiple independent, relatively small voice conferences. For example, a single media processor might divide its total voice channel processing capability between three small, but independent, voice conferences. However, such configurations led to yet another difficulty, namely resource fragmentation.

[006] Whenever a media processor hosted one or more voice conferences, each voice conference consumed a certain number of voice channel resources. As a result, a request for a new voice conference with more participants than available voice channel resources had to be refused. For example, a media processor supporting 20 voice channels, currently hosting a marketing voice conference with 10 channels and a design voice conference with 5 channels, could not support a sales voice conference requiring 6 or more channels. The remaining 5 voice channels were fragmented away from the original 20 voice channels, and were effectively an unavailable resource for the media processor.

[007] In order to expand capacity, multiple media processors were sometimes provided, with each media processor again handling the entirety of one or more voice conferences. However, even when multiple media processors were present, the IP Telephony system assigned voice conferences to the media processors in the same way. Consequently, rather than generating resource fragmentation on a single media processor, the IP Telephone system generated resource fragmentation on multiple media processors.

SUMMARY

[008] A conferencing system assigns voice conferences across multiple media processors. The conferencing system thereby allows voice conferences to proceed, even when any single media processor in the conferencing system could not support the voice conference. The conferencing system pools the voice channel resources of multiple media

processors to support more conferences, at the same time significantly reducing resource fragmentation among the media processors. The voice conferencing system may enhance business communication possibilities, without significantly increasing cost or equipment requirements.

[009] Accordingly, a voice conferencing system includes a group of media processors assigned to concurrently support a voice conference. In addition, the voice conferencing system includes distribution circuitry connected to the group of media processors. The distribution circuitry, which may be an IP router, receives data transmitted to a network distribution address, such as a multicast address, by the individual media processors. Subsequently, the distribution circuitry distributes the data received, for example, from a first media processor in the group to the remaining media processors in the group. The media processors thereby share their voice channel data with each media processor concurrently handling the voice conference.

[010] In terms of the operation of the voice conferencing system, a first media processor receives first endpoint traffic. The first media processor then transmits a selected portion of the first endpoint traffic to the distribution circuitry for distribution to other media processors. A second media processor receives second endpoint traffic, as well as the selected portion of the first endpoint traffic. The second media processor then proceeds to determine a net traffic result from the selected portion of the first endpoint traffic, as well as the second endpoint traffic.

[011] A media processor in the voice conferencing system includes a network interface that receives incoming voice conference traffic. The media processor also includes a processing unit that directs a selected portion of the incoming voice conference traffic through the network interface to a multicast network address.

[012] In operation of the media processor, the media processor first receives incoming voice conference traffic. Subsequently, the media processor selects a distribution portion of the incoming voice conference traffic. One selected, the media processor may then transmit the distribution portion to a network distribution address. The media processors thereby

3

distribute their voice channel data to each other media processor concurrently handling the voice conference.

[013] The present invention is defined by the following claims, and nothing in this section should be taken as a limitation on those claims. Further aspects and advantages of the invention are discussed below in conjunction with the preferred embodiments. Any one or more of the above described aspects or aspects described below may be used independently or in combination with other aspects herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[014] Figure 1 illustrates one implementation of a voice conferencing system that distributes a voice conference over multiple media processors.

[015] Figure 2 illustrates one implementation of a media processor that may be employed in the voice conferencing system shown in Figure 1.

[016] Figure 3 illustrates one implementation of a multipoint controller that may be employed in the voice conferencing system shown in Figure 1.

[017] Figure 4 shows one example of a signal flow diagram of voice conference traffic between the media processors, the multicast switch, and the endpoints in the voice conferencing system shown in Figure 1.

[018] Figure 5 illustrates one example of a flow diagram of the acts that a media processor may take to distribute selected incoming voice conference data to other media processors in the voice conferencing system shown in Figure 1.

DETAILED DESCRIPTION

[019] The elements illustrated in the Figures interoperate as explained in more detail below. Before setting forth the detailed explanation, however, it is noted that all of the discussion below, regardless of the particular implementation being described, is exemplary in nature, rather than limiting. For example, although selected aspects, features, or components of the

implementations are depicted as being stored in memories, all or part of systems and methods consistent with the distributed voice conferencing may be stored on or read from other machine-readable media, for example, secondary storage devices such as hard disks, floppy disks, and CD-ROMs; a signal received from a network; or other forms of ROM or RAM either currently known or later developed.

[020]   Furthermore, although specific components of the voice conferencing systems will be described, methods, systems, and articles of manufacture consistent with the voice conferencing systems may include additional or different components.. For example, a processor may be implemented as a microprocessor, microcontroller, application specific integrated circuit (ASIC), discrete logic, or a combination of other types of circuits acting as explained above. Similarly, memories may be DRAM, SRAM, Flash or any other type of memory. Databases, tables, and other data structures may be separately stored and managed, incorporated into a single memory or database, or generally logically and physically organized in many different ways. The programs discussed below may be parts of a single program, separate programs, or distributed across several memories and processors.

[021]   Figure 1 shows a voice conferencing system 100. The conferencing system 100 includes a first media processor (MP) 102, a second MP 104, and a third MP 106. The three MPs 102-106 are part of an MP group 107. The conferencing system 100 further includes a multipoint controller (MC) 108, and a multicast switch 110. An internal network 112 connects the MPs 102-108, MC 108, and the multicast switch 110.

[022]   Each MP is assigned to handle voice conference traffic for one or more endpoints. As shown in Figure 1, the first MP 102 handles the endpoints EP1-1 through EP1-r, the second MP 104 handles the endpoints EP2-1 through EP2-s, and the third MP 106 handles the endpoints EP3-1 through EP3-t. Each endpoint may communicate with the conferencing system 100 through an external network, for example, the external network 114. The endpoint may then communicate with the media processor through

5

an MP connection, for example the MP connection 116, and with the multipoint controller 108 through an MC connection, for example, the MC connection 118. Either of the MP connection 116 and the MC connection 118 may include a network address, network address and port number, or another type of network identifying information.

[023]   Although Figure 1 shows three MPs 102-106, the conferencing system 100 may include more or fewer MPs. Accordingly, additional MPs may be added to expand the overall voice conferencing capabilities of the conferencing system 100. For example, as shown in Figure 1, the MP A and MP B are present and part of the conferencing system 100, and stand ready to support an ongoing voice conference or a new voice conference. As will be explained in more detail below, the MC 108 distributes a voice conference over multiple MPs.

[024]   To that end, the MC 108 communicates with the MPs 102-106 over the internal network 112. The networks 112, 114 may adhere to one or more network topologies and technologies. For example, the networks 112, 114 may be an Ethernet network, but in other implementations may alternatively be implemented as a Fiber Distributed Data Interconnect (FDDI) network, Copper Distributed Data Interface (CDDI) network, or another network technology.

[025]   In one implementation, the networks 112, 114 are IP packet switched networks, employing addressed packet communication. For example, the networks 112, 114 may support transmission and reception of User Datagram Protocol (UDP) packets for communication between the MC 108, MPs 102-106, endpoints, and the switch 110. Other packet types may be employed however, depending on the desired underlying network implementation.

[026]   The MC 108 tracks the resource availability at each MP 102-106. For example, the MC 108 may monitor the estimated remaining voice channel capacity at each MP 102-106. The MC may then distribute endpoints in a voice conference among the MPs 102-106 in order to support a voice conference that is otherwise too large for any single MP to currently handle.

6

[027] The endpoints represent any participant in the voice conference. An endpoint is not limited to a human speaker sitting at a desk or in a conference room, however. Rather, the endpoint may represent any connection to the voice conference, including those that are automatic or mechanical in nature. For example, an endpoint may be a computer system converting speech signals to text data for later retrieval.

[028] Each endpoint communicates with the conferencing system 100 through a network, such as the external network 114. The networks generally represents a transport mechanism or interconnection of multiple transport mechanisms for voice conference traffic to and from the endpoint. As one example, the endpoint may be a home personal computer communicating over a dial-up modem, DSL, T1, or other network connection to the conferencing system 100.

[029] A conference participant at home or in an office may, for example, employ their personal computer, telephone set, or another input device, to digitize voice data received through a microphone, encode the voice data, and transmit the voice data through the external network 114 to the conferencing system 100. Similarly, the home or office computer may receive voice conference traffic through the external network 114, decode the voice data in the conference traffic, and reproduce the voice data using a sound card and speakers attached to the personal computer. Each endpoint may be assigned a network address that serves to identify the endpoint. The network address may include an IP address, for example, or an IP address and a port number. As indicated above, however, alternative addressing techniques may additionally or alternatively be employed to identify the endpoints.

[030] Any endpoint may employ multiple connections to the conferencing system 100. Consequently, an endpoint may directly communicate with the MPs 102-106 through MP connections, and also directly communicate with the MC 108 through an MC connection. To that end, each MP 102-106 and the MC 108 may include one or more dedicated network addresses and port numbers that identify the MPs 102-106 and MC 108. As examples, the network addresses may be class A, B, C, D, or E IP addresses. However, the

network addresses may adhere to other network address standards, such as the IP v 6 standard, or another network address standard. In other implementations, a single connection is provided between an endpoint and the system 100.

[031] In one implementation, the conferencing system 100 transmits and receives voice conference traffic using a high speed protocol. For example, the conferencing system 100 may employ the Real Time Protocol (RTP) over UDP to provide a responsive voice conference experience for the endpoints. In addition, the signaling between the conferencing system 100 and the endpoints may proceed according to the H.323 packet-based multimedia communications system standard published by the International Telecommunications Union (ITU). In other implementations, however, the conferencing system 100 may employ additional or alternative protocols selected according to any desired network implementation specifications. For example, the conferencing system 100 and endpoints may employ the Session Initiation Protocol (SIP) developed for Internet conferencing, telephony, presence, events notification and instant messaging.

[032] The conferencing system 100 may packetize voice conference data sent to any endpoint, or receive packetized voice conference data from any endpoint. As one example, the conferencing system 100 may distribute outgoing voice conference data into packets that contain approximately 30 ms of voice data. Similarly, the voice conferencing system 100 receive and buffer incoming voice conference data distributed among packets holding approximately 30 ms of voice data. In other implementations, however, more or less than 30 ms of voice data may be stored in each packet.

[033] As shown in Figure 1, a voice conference is in place, distributed between the three MPs 102-106 in the MP group 107. The first MP 102 processes the voice conference traffic for the 'r' endpoints EP1-1 through EP1-r. The second MP 104 processes the voice conference traffic for 's' endpoints EP2-1 through EP2-s. Similarly, the third MP 106 processes the voice conference traffic for 't' endpoints EP3-1 through EP3-t. Accordingly, the three MPs 102-106 support a voice conference with 'm' = 'r' + 's' + 't' total

8

voice channels. A voice conference may expand or contract during its existence as new endpoints join the conference, or as existing endpoints leave the conference. Consequently, the total number of endpoints may vary extensively during a voice conference. Furthermore, any MP 102-106 may belong to one or more MP groups, depending on the distribution of voice conferences between the MPs 102-106.

[034]   Figure 2 shows one implementation of a media processor 200. The media processor 200 may be implemented as a stand alone processing system, for example, or may be integrated with other processing systems present in the conferencing system 100. Each media processor in the conferencing system 100 may be implemented in the same or in a different manner than that discussed below with regard to Figure 2.

[035]   The media processor 200 includes one or more central processing units, for example, the CPUs 202, 204, 206, and 208, a network interface 210, and a network address 212 assigned to the network interface 210. In addition, the media processor 200 includes a memory 214 that may store programs or data, a conference buffer 216, and an endpoint buffer 218. The program memory may include, as examples, voice Coders / Decoders (CODECs) 220, a channel filter 222, and a net traffic filter 224. The endpoint buffer 218 is physically or logically allocated into individual buffers for each endpoint handled by the media processor 200. Figure 2 shows the EP1-1 buffer 226 and the EP1-r buffer 228 as examples.

[036]   In operation, the network interface 210 receives voice conference traffic from the endpoints. The voice conference traffic is typically encoded digitized voice samples, transmitted in UDP packets forming a voice channel to the media processor 200. A voice channel is the data flow supported by a transport mechanism between an endpoint and the media processor 200. The voice channels are implemented, for example, through unidirectional or bi-directional IP packet transmission of voice conference data from any endpoint to the media processor 200 and from the media processor 200 to the endpoint.

9

[037]   The media processor 200 stores incoming voice conference traffic from a given endpoint in an associated endpoint buffer.   In one implementation, the endpoint buffers 218 store approximately 1 - 2 packets or 20 - 50 ms of voice conference traffic, and thereby help reduce the undesirable effects of network jitter on the voice conference. The individual buffers may be enlarged or reduced however, to accommodate more or less network jitter, or to meet other implementation specifications.

[038]   As voice conference traffic arrives, the media processor 200 distributes the processing load among the data processors 202-208. The data processors 202-208 retrieve voice conference traffic from the endpoint buffers 218, and decode the voice channels in the voice conference traffic. The data processors 202-208 may apply the channel data in the voice conference traffic to the CODECs 220 to recover the digitized voice samples in each voice channel.

[039]   As the data processors 202-208 decode the voice channels, the data processors 202-208 prepare to distribute a selected portion of the voice channels to the other media processors 102-106 in the conferencing system 100. In one implementation, the media processors 102-106 apply the channel filter 222 to the voice channels in order to determine the portion of the voice channels to transmit to the other media processors 102-106.

[040]   As one example, the channel filter 222 may be an n-loudest analysis program that analyzes the decoded voice channel data to determine the 'n' loudest voice channels among the voice channels. Alternatively, the channel filter 222 may be a hardware circuit that performs the same or a different filtering function. The channel filter 222 is not limited to an 'n' loudest filter, however.   Instead, the channel filter 222 (whether implemented in hardware or software) may instead select any set of the incoming voice channels as the portion of the voice channels for distribution according to any other desired criteria. For example, the channel filter 222 may select all incoming channels, already mixed, for distribution.

[041]   Once determined, the media processor 200 transmits the voice channel data in the selected voice channels to each of the remaining media

processors in the MP group 107 that is concurrently supporting the voice conference. Accordingly, the media processor 200 may packetize and transmit the selected voice channels to the multicast switch 110. When UDP packets are employed, for example, the media processor 200 may transmit the selected voice channels to a UDP multicast address that incorporates the group address or identifier.

[042] In turn, the multicast switch 110 receives the voice channel data from the selected voice channels, and transmits the channel data to other media processors, for example, each remaining media processor. In that regard, the multicast switch 110 may determine the assigned network addresses for each remaining media processor by consulting an internal routing table. As a result, each media processor concurrently supporting a voice conference receives selected voice channels from each remaining media processor also supporting the same voice conference.

[043] The multicast switch 110 is one example of distribution circuitry that may forward the voice channel data to each MP. Other distribution circuitry may also be employed, however. As examples, the distribution circuitry may instead be a network hub or other network device that forwards packets to multiple destinations in a broadcast, multicast, or direct communication manner. Alternatively, the media processor may consult a routing table and route the channel data to other media processors without the multicast switch 110.

[044] With reference again to Figure 1, and assuming, for example, that each MP 102-106 employs an 'n' loudest channel filter 222, then the MP 102 forwards the channel data for the 'n' loudest voice channels of the voice conference traffic from EP1-1 through EP1-r to both the MP 104 and MP 106. Similarly, the MP 104 forwards the channel data for the 'n' loudest voice channels of the voice conference traffic from EP2-1 through EP2-s to both the MP 102 and MP 106. In addition, the MP 106 forwards the channel data for the 'n' loudest voice channels of the voice conference traffic from EP3-1 through EP3-s to both the MP 102 and the MP 104. The conference buffer

216 in each MP may store the received voice channels for processing by the data processors 202-208.

[045]   Each MP 102-106 therefore receives voice channel data for 'n' selected voice channels from each other MP in the MP group 107. Accordingly, each MP 102-106 obtains 3n sets of voice channel data that are the loudest among all the conference endpoints. In one implementation, the MPs 102-106 individually apply a net traffic filter 224 to the obtained 3n voice channels to determine a net traffic result to be sent back to each endpoint handled by that MP.

[046]   As one example, the net traffic filter may also be an 'n' loudest analysis program. In that case, the net traffic filter 224 in each MP 102-106 identifies the 'n' loudest voice channels from among the 3n loudest voice channels. In other implementations, however, the net traffic filter may apply different filtering criteria to the received voice channel data to select any subset of the received voice channels as the net traffic result. Furthermore, the application of the net traffic filter 222 is optional, and an MP may therefore instead send back all of the voice channels received from the remaining MPs in the MP group 107. In other words, the net traffic result may be the sum of all the selected voice channels obtained from each MP in the MP group 107.

[047]   Once the MP 102, for example, has determined the net traffic result, the MP 102 may then apply one or more CODECs 220 to individually encode the voice channels for delivery to the endpoints EP1-1 through EP1-r. Once encoded, the MP 102 delivers the net traffic result to each endpoint through the network interface 210. In that regard, the MP 102 may transmit the net traffic result via RTP over UDP to each endpoint.

[048]   As a result, the voice conference is distributed over multiple media processors 102-106. By employing the multicast switch 110, only a single transmission delay 'X' is incurred for communication between all the MPs in a MP group. Assuming each MP takes 'Y' time to process the voice conference traffic, then the total delay for distributed voice conferencing is only X+Y. Because X and Y may each be under 20 ms, the total delay may be under 40 ms.

12

[049] The delay 'X' is independent of the number of media processors in an MP group. As a result, even when additional MPs are added to support an ongoing voice conference, the total delay remains X+Y. A voice conference may dynamically grow or shrink without adverse delay impacts on the conference participants.

[050] The voice conferencing system 100 decentralizes voice conference processing from a single MP to multiple MPs in an MP group. Nevertheless, the MP group may physically reside at a centralized location and remain part of a centralized voice conferencing system. The voice conferencing system 100 may thereby represent a centralized conferencing approach with internal decentralization.

[051] Although the voice conferencing system 100 may be decentralized, the delay 'X' does not increase as the conferencing system 100 distributes a voice conference over multiple MPs. Accordingly, whether a voice conference starts in a distributed manner over multiple MPs, or grows to span multiple MPs, the conference participants do not experience reduced voice conference quality from decentralization. Instead, the participants may encounter a consistent voice conference experience, even as the conference grows or shrinks across more or fewer MPs.

[052] While multicasting the voice channel data between media processors has certain advantages, it is not the only way to distribute the voice channel data. Rather, the media processors may employ any desired communication mechanism for sharing their selected voice channels between the remaining media processors. For example, the media processors may sequentially transfer voice channel data through direct communication with each media processor.

[053] Figure 3 illustrates a multipoint controller (MC) 300 that may be employed in the conferencing system 100. The MC 300 includes a processor 302, a network interface 304, and a network address 306 assigned to the MC 300. A memory 308 in the MC 300 includes a channel capacity table 310. The channel capacity table 310 includes a media processor field 312 and an estimated remaining channel capacity field 314.

13

[054]   In the example shown in Figure 3, the channel capacity table 310 includes a media processor field entry for each of the MPs 102-106, and well as for a fourth MP labeled D. Associated with each media processor field entry is an estimated remaining channel capacity. As shown, the MP 102 has the capability to handle 5 additional voice channels, the MP 104 has the capability to handle 10 additional voice channels, and the MP 106 has the capability to handle 5 additional voice channels. The MP A in the voice conferencing system 100 has the capacity to handle 10 additional voice channels, while MP B has no remaining capacity.

[055]   The MC 300 maintains the channel capacity table 310 through, for example, periodic communication with the media processors in a conferencing system. Thus, the media processors may report their estimated remaining channel capacity to the MC 300 at selected times, intervals, or periods. Additionally or alternatively, the MC 300 may be pre-configured with the total estimate channel capacity of each media processor, and may then maintain the channel capacity table 310 based on assignments and releases of endpoints to and from media processors as explained below. Additionally or alternatively, the MC 300 may track channel capacity at each MP in other ways or using a different table structure or data structure.

[056]   The endpoints may communicate directly (or indirectly via a media processor) with the MC 300 through the network interface 304. As examples, the endpoints may request to join a voice conference, or inform the MC 300 that the endpoint is leaving an existing voice conference through an MC connection 118. In response, the MC 300 determines which media processor to assign to the voice conference, in keeping with the estimated channel capacities available at each media processor.

[057]   The MC 300 may allocate the endpoints to the media processors in many different ways. For example, assuming that the MC 300 will setup a new voice conference with 20 voice channels, there is no single MP that can handle the voice conference. Without distributing the new voice conference among the existing MPs, the total unused channel capacity of 30 voice channels would be wasted. However, the distributed conferencing system

14

100, through the communication techniques described above, treats all the available voice channel capacity among disparate media processors as a single logical pool of voice channel resources.

[058]   Consequently, the MC 300 selects two or more media processors to concurrently handle the new voice conference. For example, the MC 300 may select the fewest number of media processors needed to handle the new voice conference. In that case, the MC 300 would select the MP 104 and the MP A to handle the new voice conference. The MP 104 and the MP A may then form a second MP group with its own unique identifier that may be used as part of a UDP multicast address for the second MP group. As other examples, the MC 300 may select the greatest number of media processors needed to handle the new voice conference, the fastest media processors, sequentially pick media processors from the channel capacity table 310, randomly pick media processors form the channel capacity table 210, or choose media processors according to any other selection technique.

[059]   After determining which media processors will handle the new voice conference, the MC 300 updates the channel capacity table 310. The MC 300 then communicates voice conference setup information over the network 112 to each selected media processor. As examples, the setup information may include the number of voice channels that the media processor will need to support for the new voice conference, the network addresses of the endpoints that the media processor will support, the group identifier that may form part of the multicast address for the media processors handling the new voice conference, the appropriate CODEC to apply for the endpoint, and the like.

[060]   Once the voice conference is established, the media processors directly handle incoming and outgoing voice conference traffic with their assigned endpoints. As new endpoints request to join a voice conference, the MC 300 may again consult the channel capacity table 310 to determine which media processor will support the new endpoint. The MC 300 responsively updates the channel capacity table 310 and communicates the setup information to the media processor. Similarly, as endpoints inform the MC

15

300 that they are dropping from the voice conference, as drops are detected or as the media processors report dropped endpoints, the MC 300 updates the channel capacity table 310.

[061] Figure 4 shows a signal flow diagram 400 that traces incoming and outgoing voice conference traffic through the voice conferencing system 100. In Figure 4, EP1 represents the endpoints EP1-1 through EP1-r, EP2 represents the endpoints EP2-1 through EP2-s, and EP3 represents the endpoints EP3-1 through EP3-t. Incoming voice conference traffic 402 arrives at the MP 102 from EP1. Similarly, incoming voice conference traffic 404 and 406 arrives at the MPs 104 and 106, respectively.

[062] Each MP 102-106 applies a channel filter to its incoming voice conference traffic. As a result, the MP 102 transmits selected voice channels 408 originating with EP1 to the multicast address, and thereby to the multicast switch 110. For example, the MP 102 may transmit the 'n' loudest voice channels to the multicast switch 110. In addition, the MP 104 applies its channel filter, selects one or more of its incoming voice channels to transmit to the multicast address, and transmits the selected voice channels 410 to the multicast switch 110. Selected voice channels 412 determined by the MP 106 also arrive at the multicast switch 110.

[063] The multicast switch 110 receives the selected voice channels 408-412 from each MP 102-106. Because the UDP packets specify a MP group address, the multicast switch 110 may consult an internal routing table to determine the assigned network addresses for the MPs in the corresponding MP group 107. The multicast switch 110 then proceeds to forward the selected voice channels form each MP every other MP in the MP group 107.

[064] More specifically, the multicast switch 110 forwards the selected voice channels 408 from MP 102 to the MP 104 and the MP 106. Similarly, the multicast switch 110 forwards the selected voice channels 410 from MP 104 to the MP 102 and the MP 106. The selected voice channels 412 from MP 106 arrive, through multicast transmission, at the MP 102 and the MP 104, as shown in Figure 4.

16

[065] Each MP 102-106 independently determines a net traffic result from all of the voice channel data received from other MPs in the MP group 107. As an example, each MP 102-106 may determine the 'n' loudest voice channels present at any given time among all the endpoints EP1-3. Subsequently, each MP 102-106 communicates the net traffic result to the endpoints assigned to that MP.

[066] Figure 4 shows that the MP 102 transmits a net traffic result as outgoing voice conference traffic 414 to the EP1. In addition, the MP 102 transmits a net traffic result as outgoing voice conference traffic 416 to the EP2. In the same manner, the MP 106 determines a net traffic result, and transmits it as the outgoing voice conference traffic 418 to the EP3.

[067] Figure 5 shows a flow diagram 500 of the acts taken by a media processor 102-106 in the distributed voice conferencing system 100. For example, the media processor 102 may first receive incoming voice conference traffic 402 from the endpoints EP1-1 through EP1-r (Act 502). The endpoint buffers 218 temporarily store the incoming voice conference traffic 402 (Act 504). Once the incoming voice conference traffic 402 has arrived, the media processor 102 may then apply one or more CODECs 220 to the voice channels in the voice conference traffic 402 to decode the digitized data samples (Act 506).

[068] With or without the voice channels decoded, the media processor 102 may apply a channel filter 222 to determine one or more voice channels in the incoming voice conference traffic to forward to other media processors (Act 508). For example, the media processor 102 may apply an n-loudest channel filter to select fewer than all voice channels from the incoming voice conference traffic. The selected voice channels are then transmitted to the remaining media processors in the media processor group 107 (Act 510). To that end, the media processor 102 may transmit the selected voice channels in UDP packets to a UDP multicast address including a group identifier for the media processor group 107.

[069] The multicast switch 110 receives the selected voice channels on the multicast address. In response, the multicast switch 110 determines the

17

assigned network addresses for each remaining media processor in the processor group 107. The multicast switch 110 then transmits the selected voice channels to each of the remaining media processors 104, 106. Each remaining media processor 104, 106 performs the same processing steps on incoming voice conference traffic 404, 406.

[070] Accordingly, the media processor 102 receives multicast transmissions of voice channel data originating with the media processors 104 and 106 (Act 512). In order to determine which voice channels to forward to the endpoints EP1-1 through EP1-r, the media processor 102 applies a net traffic filter to the voice channel data received in addition to its own voice channel data (Act 514). Thus, for example, although the media processor 102 may obtain 3n loudest voice channels, the media processor 102 selects, for example, 'n' loudest of the 3n loudest voice channels as the net traffic result. The media processor 102 thereby may keep the conference participants from becoming overwhelmed with information.

[071] Having determined the net traffic result, the media processor 102 mixes each channel in the net traffic result into an output stream. The media processor 102 may then apply one or more of the CODECs 220 to the output stream. The media processor 102 thereby encodes the net traffic result for each endpoint according to the CODEC previously negotiated for that endpoint (Act 516). The media processor 102 may then forward the net traffic result in the form of encoded output streams to its endpoints EP1-1 through EP1-r (Act 518). The media processor 102 determines whether any endpoints are still participating in the voice conference (Act 520). If so, processing continues as noted above. Otherwise, the media processor 102 may terminate processing.

[072] The distributed conferencing system 100 assigns a single voice conference over multiple media processors. As a result, the conferencing system 100 is not limited to running any given voice conference on a single media processor. Even though each media processor has a finite channel capacity, the conferencing system 100 may allow additional voice conferences to proceed by pooling resources from multiple media processors.

18

As an additional benefit, the conferencing system 100 may experience less resource fragmentation than prior systems. In other words, the conferencing system 100 may more efficiently employ the hardware already present in the conferencing system 100 to support more voice conferences than would be possible otherwise.

[073] It is therefore intended that the foregoing detailed description be regarded as illustrative rather than limiting, and that it be understood that it is the following claims, including all equivalents, that are intended to define the spirit and scope of this invention.